

EXHIBIT 28

Google Groups

Persistence Framework

mzomer

Mar 8, 1996 3:00 AM

Posted in group: **comp.lang.smalltalk**

In article <4hlfr\$f...@steel.interlog.com>, cfw...@interlog.com says...

>

>mzo...@inter.nl.net (mzomer) wrote:

>

>>Hi,

>

>

>>I am looking for a good Persistence Framework that does not have any
>>implications on existing code. We use Team/V 3.0.1 from Digitalk (on
>>a OS/2 platform) and store our business objects in a DB2/2 or Oracle
>>database. Before building our own Framework I want to hear from you!

>

>

>>CU

>

>>--martin

>

>I would recommend TOPLink from The Object People

>Contact pa...@objectpeople.com

>

>CF Wong

>Bank of Montreal

>

>

Thanks for your reply,

I received a white paper describing the TOPLink product including some coding examples. I did not like the way they solve the problem.

1. The product generates code which is added to the business objects. If you want to switch to an OODBMS you have a major problem.
2. The product also enforces business object to have a KEY, sorry, but that not Object Oriented at all.
In a RDBMS you need a key to find related objects (rows).
In a Object Oriented language you have references (pointers) to related objects. The way you navigate through the business object model is via references and in a RDBMS via associations.

A way to solve the mapping from a business object model to a RDBMS is using a Meta Model which describes your business object model including relations. Smalltalk itself does not have explicit support for relations. Using the Meta Model and a description in Smalltalk of the database structure (this is also a Meta Model) you are able to map object ID to keys.

The code you need to store or retrieve objects can be generic, so you do not need to generate code. Generating code is for those programmers who are not able to write generic solutions. If you have the knowledge to write a code generator you also have all the knowledge to write a generic solution.

CU

--martin

Google Groups

Persistency FrameWork

Alan Knight

Mar 8, 1996 3:00 AM

Posted in group: **comp.lang.smalltalk**

In <4hlfqr\$f...@steel.interlog.com> cfw...@interlog.com (C.F. Wong) writes:

>I would recommend TOPLink from The Object People

>Contact pa...@objectpeople.com

>CF Wong

>Bank of Montreal

Thanks for the recommendation. I think the e-mail address should be
pa...@objectpeople.on.ca. We also have a web page at
<http://www.objectpeople.on.ca> which has information on TOPLink.

--

Alan Knight	The Object People
kni...@acm.org	Smalltalk and OO Training and Consulting
al...@objectpeople.on.ca	509-885 Meadowlands Dr.
+1 613 225 8812	Ottawa, Canada, K2C 3N2

Google Groups

Persistency FrameWork

Alan Knight

Mar 8, 1996 3:00 AM

Posted in group: **comp.lang.smalltalk**

In <Dnx11...@solair1.inter.NL.net> mzo...@inter.nl.net (mzomer) writes:

>I received a white paper describing the TOPLink product
>including some coding examples. I did not like the way
>they solve the problem.

- > 1. The product generates code which is added to the business
- > objects. If you want to switch to an OODBMS
- > you have a major problem.

I'm not sure how you got this impression, but it's not true. TOPLink requires you to write a "descriptor" for each persistent class. This is essentially the meta-model you talk about below. This descriptor is typically written as a class method, but this is only because it's a convenient place to put it. We have clients who read these descriptors from a database on system startup, so they end up not being image-resident at all.

- > 2. The product also enforces business object to have a KEY,
- > sorry, but that not Object Oriented at all.
- > In a RDBMS you need a key to find related objects (rows).
- > In a Object Oriented language you have references (pointers)
- > to related objects. The way you navigate through the
- > business object model is via references and in a RDBMS via
- > associations.

No, it's not object-oriented. It's relational. If I have an object in memory that corresponds to one or more relational rows, I need a way to find which rows to update. That means I need a key. The key can be completely arbitrary (it's often generated by the database) and the application can ignore it, but it's the translation into relational terms of object identity. I guess we do affect the business objects in that the key is generally stored as an instance variable, but I would characterize this as minimal.

- >A way to solve the mapping from a business object model to a
- >RDBMS is using a Meta Model which describes your business object
- >model including relations. Smalltalk itself does not have explicit
- >support for relations. Using the Meta Model and a description in
- >Smalltalk of the database structure (this is also a Meta Model)
- >your able to map object ID to keys.

How do you propose to do this mapping from object ID to keys? Object identity is a lovely property, but it doesn't easily generate something that can be stored in a relational database.

- >The code you need to store or retrieve objects can be generic,
- >so you do not need to generate code.

And we don't.

It's possible that when you talk about generating code you mean our descriptor generator. This is not generating code for the domain objects, it is generating the meta-model automatically by examining the objects.

I think you have misconstrued how TOPLink operates. If you contact us I'm sure we can clear up any misunderstandings. If you don't like TOPLink, that's fine, but I'd hate to have you reject it as a solution based on a misunderstanding.

--

Alan Knight | The Object People
kni...@acm.org | Smalltalk and OO Training and Consulting
al...@objectpeople.on.ca | 509-885 Meadowlands Dr.
+1 613 225 8812 | Ottawa, Canada, K2C 3N2

Google Groups

Persistency FrameWork

mzomer

Mar 12, 1996 3:00 AM

Posted in group: **comp.lang.smalltalk**

From mzo...@inter.nl.net Mon 11 Mar 96 22:36:14
Xref: sun4nl comp.lang.smalltalk:31714
Path:
sun4nl!surfnnet.nl!swsbe6.switch.ch!swidir.switch.ch!in2p3.fr!univ-
lyon1.fr!fdn.fr!r2d2.fdn.org!uunet!in1.uu.net!newsfeed.int
ernetmci.com!news.kei.co
From: kni...@mrco.carleton.ca (Alan Knight)
Newsgroups: comp.lang.smalltalk
Subject: Re: Persistency FrameWork
Date: 10 Mar 96 17:45:47 GMT
Organization: The Object People
Lines: 152
Message-ID: <knight.8...@tina.mrco.carleton.ca>
References: <DnpM7...@solair1.inter.NL.net> <4hlfqr\$f...@steel.interlog.com>
<Dnx11...@solair1.inter.NL.net>
<knight.8...@tina.mrco.carleton.ca> <
Reply-To: kni...@mrco.carleton.ca (Alan Knight)
NNTP-Posting-Host: tina.mrco.carleton.ca

>>It might be convenient to add the description as a class method to
>>the business object but it is terrible wrong!

>If you think it's wrong, don't do it. I personally don't consider that
>an intrusive thing in the business object, and I think it's actually
>easier to maintain that way, but if you don't like it, you don't have
>to do it. TOPLink doesn't require the descriptor to be created from a
>class method. All it requires is that for each class to be made
>persistent, a descriptor for that class is registered with the
>database session. You can put code for them in a different class, read
>descriptor objects from the database, or load them off your Microsoft
>Watch (tm). TOPLink doesn't care.

I thought that TOPLink required a descriptor as class method but if
it is possible to put it somewhere else than this problem is
solved. I do not like the idea that external interfaces have impact
on my business object model. In many cases you have more than one
external interface and if they all require some methods to be added to
the business object model than it can create a lot of pollution.

I discussed this issue with a colleague and we came to the
same conclusion that it is possible to move the descriptors to
somewhere else.

In our applications we use a layered application structure and
database mapping do not belong to the Problem Domain Layer in
which the business object model is defined.

>>The fact that you need a key to identify the persistent object in the

>>database is right, but this piece of information can also be stored
>>somewhere else. As you say: "The key can be completely arbitrary (it's
>> often generated by the database)", is also a good reason to sepperate
>>this information from the business object.

>You don't always want to separate it. The key is often a useful piece
>of domain information, even if it is generated. For example, in an
>insurance application I've been involved with, the policy number is
>automatically created from a sequence in the database, but it is
>nevertheless important domain information. However, let's assume that
>in a particular case we want a complete separation. This is easy to
>do, as described below.

You're right, in some cases a key has also meaning in a business object,
but than and only than there will be a mapping defined. I think we
solved problem number two.

>>When you retrieve a row from the database you use the static map
>>and the Meta Model to instantiate an object. At that point you
>>have references to both worlds in one hand, so put this information
>>into the dynamic map. If you want to delete or update an object you
>>use the dynamic map to find the corresponding row in the database.
>>The static map is used to translate instance variables to columns.

>I would say that in this case each object has a key, it's just that
>you have arranged the way the key is stored slightly differently. If
>you prefer this way of doing it, this is trivial to implement in
>TOPLink. Instead of using a DirectToFieldMapping for the primary
>key(s), which map to an instance variable(s), use a
>TransformationMapping. This allows you to write arbitrary code to map
>between the database field and its representation in memory. This can
>do the lookup in the dynamic map.

Does TOPLink allow you to solve the key mapping once and for all, i.e,
write the abitray code only once?

>If there is no way to definitely and uniquely identify a row in the
>database, then you will have trouble mapping that to an object model.
>This is a fundamental problem, and certainly not unique to TOPLink. I
>don't believe TOPLink has any stronger requirement than this as far as
>database layout. We have worked with numerous clients who have
>existing databases whose schemas are not subject to modification.

You misunderstood me, in our databases each row in a table can
be uniquely identified, so that is not the problem. The problem is
that sometimes we are not allowed to make any change to a
database because it already exists. Adding our own tables is
out of the question.

>We have a (new in version 2.0) tool that can help people by generating
>the meta-model for the most common cases . You don't have to use it.
>It can put the methods anywhere.

Do you mean TOPLink has a browser which allows you to map objects
to relational tables? That's nice, most mappings are triavial so
why coding them by hand.

>I also consider that there is an enormous difference between

>generating code in a business object and generating descriptor code.

I agree, I've seen many generators which generate plain smalltalk code were the code generator is also written in smalltalk. In rare cases you might need that but most of the time you can solve your problem in a more generic way.

>The former implies making the object responsible for its own
>persistence by executing framework code. The code for the descriptor
>is only initialization code for the meta-model. The domain objects
>never execute code to read and write from the database.

Yep, I saw the examples in the white paper, so the code generator problem is also solved.

>You misunderstood the code generation issue.

Using the term 'code generator' can be very misleading. TOPLink does not generate real code it's only a description written in smalltalk.

>I think you misunderstand
>the separation issue as well. I consider the modifications TOPLink
>normally requires to business objects extremely minor and definitely
>not in the category of "screwing up".

Read my comment on more than just one external interface. I hope you interpreted the "screwing up" right :-)

>However, if you strongly object
>to the addition of one class method and (sometimes) an additional
>instance variable for the key, you can very easily eliminate both of
>them, leaving your domain objects absolutely unmodified. I repeat:
>TOPLink will work just fine without ANY modifications to your domain
>objects whatsoever.

That's all I wanted to hear from you, I must be in a good mood being so nice this time :-)

>This is exactly what TOPLink does. You say
>
>session writeObject: anEmployee
>
>and to read
>
>session
> readObjectOfClass: Employee
> where: [:emp | emp manager address streetName = emp address streetName].

OK, that's also a thing that I wanted to hear from you. I assume there is also a readObjectOfClass without a where clause?

>>I hope I made clear that making objects persistent is
>>more than just adding some code to the objects itself.
>>You have to build a real Framework that sepperates
>>interface from implementation.

>I believe we have done exactly that, and that TOPLink is extremely
>close to what you have described. The differences you talk about are
>extremely small, and as I have described above, very easy to overcome.

It took a long way to get here, I strongly recommend to extend the white paper with some explanation that one can customize TOPLink in such a way that it has no impact on the business object model.

>-----

>With reference to your posting replying to Kyle Brown, you write:

>>I do know now, and really it is not a meta-structure at all. It only
>>is a hard-wired mapping from the object world to a RDBMS.

>What's hard-wired about it? You can change descriptors very easily,
>and that include changing them while connected to a database. Under
>normal circumstances one doesn't do that, but it's entirely possible.

Hard-wired is perhaps the wrong word. I mean that TOPLink does not have separate descriptions for the business object model and the database and a mapping in between, it is all done in one description.

In our project we are thinking on how to solve the persistency framework problem. As soon as you start thinking about this problem you discover that Smalltalk has no build in support for relationships. And you need to know it, else you won't be able to store anything in a RDBMS. Another problem we want to solve is lazy object instantiation, you do not want that one read operation results in reading the complete database into memory. You can use meta level programming to solve the lazy object instantiation problem and more. For example in Smalltalk you do not know the "type" of an instance variable until you assign one. If you want to map a TABLE to an object, sometimes you need to know more than that a database type CHAR maps to a Smalltalk class String.

To solve the problems described above we came up with the following solution:

1. We need a description of our business object model that also defines relationships and attributes (incl. types).
2. We need a description of the structure of the database (Table, Column, Constraint, Key, etc.)
3. We need some clever mapping between these two worlds.
4. We need an Archiver that does the store/retrieve and holds the mappings.
5. We need a mechanism for lazy object instantiation implemented by using meta level programming.

To start with the last one, in our solution business objects do not have explicit instance variables (they are defined in the meta class) although a business object does have values (a so called valueDictionary).

We use the doesNotUnderstand message to resolve references to

instance variables (also to related objects).

The doesNotUnderstand uses the meta class to verify if an instance of an business object understands the selector for an instance variable (attribute). It also detects if the value of the instance variable is empty. If the value is empty it will retrieve it from the database.

The meta class model for the business object model and the database are trivial so I won't discuss them here.

The mapping is very similar to the one TOPLink uses.

We are also thinking of adding some explicit state to the business objects, i.e., Empty, Deleted etc. These are some ideas we are thinking of. Some parts are implemented to check if the concept is working and it does. The mapping combined with the algorithms for storing, retrieving and deleting is the most difficult one.

>>Again I disagree, TOPLink is NOT a persistency Framework, it can store >>object into a RDBMS and that's it!

>I'm not sure I understand this comment. TOPLink reads and writes >objects to and from relational databases. What else is a relational >persistency framework supposed to do?

What I mean is that TOPLink is only a persistency framework for storing/retrieving object from a RDBMS. When I talk about a persistency framework I mean a framework for storing/retrieving objects in general, so one protocol for all kinds of implementations where a relational persistency framework is one of them.

Conclusion:

A closer look to TOPLink might be useful. The white paper can be misleading in some cases and needs some rewriting and some additional examples.

It is more flexible than I thought after the discussion with Alan.

We stil disagree on sepparating the descriptions/mappings from the business object model, but if you can have it both ways than there is not really a problem except that I do not think that is clean Object Oriented if you mix two worlds. There must be something in between that knows all of both, and really guys you are so close to that.

Question:

Does TOPLink also ships the source code (or parts) with the product. To put it in other words, can it be customized?

CU

--martin

Google Groups

Persistency Framework

Alan Knight

Mar 14, 1996 3:00 AM

Posted in group: **comp.lang.smalltalk**

I think this discussion has gone beyond the point of general technical interest, so I've taken it to e-mail. If anyone's particularly interested in the details of the discussion, let me know.

--

Alan Knight | The Object People
kni...@acm.org | Smalltalk and OO Training and Consulting
al...@objectpeople.on.ca | 509-885 Meadowlands Dr.
+1 613 225 8812 | Ottawa, Canada, K2C 3N2